

Patent  
Attorney's Docket No. P2248-472

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**



In re Patent Application of )

Gordon R. MEYER et al. )

Application No.: 09/074,544 )

Filed: May 8, 1998 )

Group Art Unit: 2173 )

Examiner: B. Huynh )

Appeal No. \_\_\_\_\_ )

For: METHOD FOR DYNAMICALLY )  
GENERATING A "TABLE OF )  
CONTENTS" VIEW OF A HTML- )  
BASED INFORMATION SYSTEM )

**RECEIVED**  
AUG 22 2001  
Technology Center 2100

**SUBSTITUTE BRIEF FOR APPELLANTS**

Assistant Commissioner for Patents  
Washington, D.C. 20231

Sir:

This appeal is from the decision of the Primary Examiner dated November 7, 2000 (Paper No. 8), finally rejecting claims 1-61, which are reproduced as an Appendix to this brief.

Two extra copies of this substitute appeal brief are being filed herewith. A check covering the \$310.00 Government fee was previously submitted with the original appeal brief filed on June 11, 2001.

The Commissioner is hereby authorized to charge any appropriate fees under 37 C.F.R. §§1.16, 1.17, and 1.21 that may be required by this paper, and to credit any overpayment, to Deposit Account No. 02-4800. This paper is submitted in triplicate.

I. Real Party in Interest

The entire interest in the present application, and the invention to which it is directed, is assigned to Apple Computer, Inc., as recorded in the Patent and Trademark Office at Reel 9384, Frame 0078.

II. Related Appeals and Interferences

The Appellants' legal representative and assignee do not know of any other appeals or interferences which will directly affect, or be directly affected by, or have a bearing on the Board's decision in this Appeal.

III. Status of Claims

The present application contains claims 1-61, all of which are currently pending. Claims 2-23 and 25-61 form the basis for this Appeal. Claims 6-15, 29-38, and 46-59 stand finally rejected under 35 U.S.C. § 102(e) as allegedly being unpatentable over DeRose et al. (U.S. Patent No. 6,055,544). Claims 2-5, 16-23, 25-28, 39-45, 60-61 stand finally rejected under 35 U.S.C. § 103(a) as allegedly being unpatentable over DeRose et al. (U.S. Patent No. 6,055,544) in view of Walls et al. (U.S. Patent No. 5,848,410).

IV. Status of Amendments

An Amendment After Final Rejection was filed March 7, 2001. The substance of the Amendment was to rewrite claims 2 and 25 in independent form to incorporate the features of claims 1 and 24, and to cancel claims 1 and 24.<sup>1</sup> However, entry of the Amendment was refused by the Examiner in the Advisory Action dated March 29, 2001 (Paper No. 12). Apparently, the Examiner considered the inclusion of the word "by", to

---

<sup>1</sup> Appellants note that in the marked-up copy of claims 2 and 25 attached to the Amendment After Final Rejection filed March 7, 2001, the notation of the additions to claims 2 and 25 was inadvertently reversed — the text of claims 2 and 25 was underlined, while the text added to claims 2 and 25 was not. However, the substance of claims 2 and 25 remains the same.

provide proper grammar when rewriting claims 2 and 25 in independent form, to introduce new issues that required further consideration or search.

V. Summary of the Invention

The present invention relates generally to on-line help systems for computers. More particularly, the present invention relates to a method for providing an up-to-date table of contents of the help topics that are currently available to a user of the computer system.

It is common practice for programmed data processing systems to provide on-line assistance to an operator or user in response to a request for help, by displaying help information to the operator at a workstation. The assistance may consist of actual help information or of a menu or index of possible help topics from which the operator can select an item of interest. [See present application, page 1, lines 7-11].

Conventional computer help systems require an author to create a table of contents that reflects the individual components that are known to exist on the computer system. In such conventional computer help systems, links to individual "books" and "chapters" (*i.e.*, subsections) are typically hard-coded to reflect the current location of the help information files. Once installed, if users move or delete files, these links are broken, causing at least a portion of the table of contents to cease to function. Additionally, if new information is added to the computer system, the pre-authored table of contents will fail to reflect the addition of the new information. [See present application, page 1, lines 12-26].

Other conventional computer help systems provide a table of contents view that is compiled by the author and that represents the contents of the computer system that are available at the time of compilation. However, these other conventional computer help systems fail to provide the user with an up-to-date listing of available computer system information which is reflective of changes made to the computer system after the compilation. [See present application, page 2, lines 4-10].

Appellants' invention is directed to a method that automatically provides a user with a table of contents that is reflective of the information that is currently available on a

computer system. The table of contents is generated by scanning specific files to identify those of a particular type. For instance, one implementation of the present invention can be used to provide a table of contents for a "Help" System in a computer. In such an implementation, the files stored on the computer (and perhaps elsewhere) are scanned to locate those which pertain to the Help System. Once located, these files are then further analyzed for pre-defined information, such as HTML meta-tags. These meta-tags indicate the information that is to be displayed in the table of contents, such as the title of a chapter in the Help System. The information gathered is merged with pre-defined HTML template files that control the formatting and presentation of the information. The resulting table of contents which is provided to the user accurately reflects the current content of the information system. [See present application, page 6, line 22 to page 7, line 26].

The present invention offers the advantages of creating a table of contents on demand, based on the current contents of the information system being accessed. The dynamically-generated table of contents is based upon information contained in numerous, individual files located within the information system. No compilation of the files that make up the system is necessary. All files are text-based, and in a standard format, such as HTML. The format of the table of contents is determined by pre-authored template files. This allows each module to have its own style of presentation. [See present application, page 7, lines 8-23].

One of the advantageous characteristics of the present invention is its use of a multi-file information system from which to dynamically generate a table of contents. In the present invention, a dynamically-generated table of contents based on a multi-file information system allows both users and developers to freely move and delete individual information files within the system without the need to re-create the table of contents. Once the system is accessed by the user or developer, any changes to information files are automatically detected and result in the table of contents being regenerated. These changes are then reflected in the regenerated table of contents. [See present application, page 15, line 19 to page 16, line 4].

VI. The Issues

The final Office Action presents two issues for review in this Appeal:

Are claims 6-15, 29-38, and 46-59 unpatentable under 35 U.S.C. § 102(e) in view of U.S. Patent No. 6,055,544 to DeRose et al. ("DeRose")?

Are claims 2-5, 16-23, 25-28, 39-45, and 60-61 unpatentable under 35 U.S.C. § 103(a) over DeRose in view of U.S. Patent No. 5,848,410 to Walls et al. ("Walls")?

VII. Grouping of Claims

Appellants do not consider all of the claims that are rejected under 35 U.S.C. § 102(e) to stand or fall together. For the reasons presented in Sections VIII(D)-(I) of this Brief, claims 8, 31, and 48 (as a group), claims 10, 33, and 50 (as a group), claims 11, 34, 51 (as a group), claims 12, 35, and 52 (as a group), claims 15, 38, and 59 (as a group), and claims 55 and 56 (as a group) present issues for patentability which are separate from those presented by the remaining claims (which can be treated as a single group for purposes of this appeal).

With respect to the claims that are rejected under 35 U.S.C. § 103(a), for purposes of expediency, all of these claims can be treated as a single group for purposes of this appeal.

VIII. Argument

A. The DeRose Patent

DeRose relates to computer systems for electronic publishing of electronic documents. DeRose notes several problems with the current structure of the World Wide Web (WWW) and the protocols used thereon. For instance, there is no protocol which allows access to only a portion of a large document. To provide acceptable performance, publishers currently maintain a large document as a collection of small document fragments, which leads to document management problems and burdens for the publishers. In addition, there is no protocol for linking to targets that are a portion of a document. Therefore, it is a stated aim of DeRose to provide a mechanism for accessing only a

portion of a large electronically published document, and to automatically determine what portion of the document to select as a previous portion or a next portion *without* maintaining separate data files of each portion of the document. [See DeRose, col. 4, lines 34-38].

To solve these and other related problems, DeRose discloses a method of using meta-tags inserted into large electronic documents to provide easier retrieval and access to selected portions of the document over remote connections. [See DeRose, col. 11, lines 27-31]. According to DeRose, through the use of meta-tags, a document is broken down into parts (e.g., books, chapters, pages, and so forth), with the accompanying titles for each of the parts. [See DeRose, col. 8, lines 18-23]. Once broken down, a user can retrieve, view, and navigate between *self-contained* fragments of a document located on a remote server. [See DeRose, col. 4, lines 34-36]. The table of contents that are created for a document by DeRose allow a user to access a given part or portion of the document. Thus, DeRose teaches a method that allows a user to retrieve portions (i.e., self-contained fragments) of an individual document.

#### B. The Walls Patent

Walls relates to data processing, search, and display systems that facilitate access to files in large databases. More particularly, Walls relates to index files in publicly- and privately-accessible networks.

Walls discloses a method and system for searching one or more files within a file system to provide the user with an updated index. According to Walls, information stored in one or more sources of information, such as a database, is continuously indexed. [See Walls, col. 3, line 64 to col. 4, line 2]. To generate the index, Walls uses data structures for storing and processing information extracted from the header lines of web pages in file systems chosen by a user. [See Walls, col. 4, lines 30-35]. Linked lists are created in these data structures to allow rapid construction and display of an alphabetical index of keywords from the header lines, with each keyword having associated with it a title extracted from the same web-page header. [See Walls, col. 5, lines 20-42]. The

alphabetical index by keywords can be displayed on a file that permits the user to jump to a desired location in the alphabetical index. Alternatively, the user may search the alphabetical index to find titles or keywords that correspond with an entered character string. Also, a user may select a title in the index and view the file from which the title was extracted. [see Walls, col. 6, lines 16-34]

C. Claims 6-7, 9, 13-14, 29-30, 32, 36-37, 46-47, 49, 53-54, 57-58

Independent claim 6 of the present application recites the step of "indexing each file and a first level of each book of a predetermined folder for *files* of a first type." Independent claims 29, 46, and 57 recite similar steps or features. DeRose does not disclose or suggest such steps or features, because DeRose is directed to an entirely different problem from the present invention, and the solution to that problem does not require or utilize such an indexing process.

DeRose teaches a method that allows a user to retrieve portions (i.e., self-contained fragments) of an individual document. [See DeRose, col. 4, lines 34-36]. Since DeRose is only concerned with the contents of a single document at a time, DeRose does not teach looking outside that single document for any additional information. In other words, since the information required by DeRose is wholly contained within an individual file, DeRose does not disclose or suggest a need for locating or retrieving information outside of the individual document. Consequently, since DeRose applies to the contents of an individual document, DeRose does not disclose or suggest the step of indexing each file and a first level of each book of a predetermined folder for *files* of a first type. In this regard, it is important to keep in mind that DeRose seeks to *avoid* the need for multiple data files. [See DeRose, col. 4, lines 37-38].

In addition, DeRose does not disclose or suggest the step of scanning the files of a first type for at least one HTML meta-tag of a predetermined type in order to determine first data to be added to a first table of contents, as also recited in claim 6. Independent claims 29, 46, and 57 recite similar steps or features. As noted previously, DeRose provides a method by which a user can access individual sections of a document. The

information required by DeRose to perform this method is completely contained within the document itself (i.e., in the embedded meta-tags). Consequently, DeRose does not disclose or suggest a need to scan for files to retrieve information residing outside of an individual file.

In the rejection of claims 6-15, 29-38, and 46-59, the Examiner has asserted that DeRose teaches a "Web site" which comprises a plurality of related HTML documents listed in a table of contents. [See Final Office Action, Paper No. 8, section 5, page 4]. It is further asserted by the Examiner that for each document, a second table of contents for the document is generated to help downloading a specific portion of the document instead of the entire document to reduce processing time. It is contended by the Examiner that Applicants appear to take the example given by DeRose at column 18, lines 22-24 as DeRose's invention and argue that DeRose's invention deals with a single document only and that Applicants' argument is "not true" when DeRose is considered as a whole.

Contrary to the Examiner's assertions, according to the express teachings of DeRose, DeRose discloses a method of demarcating a single document to permit a user to view self-contained document fragments. Consequently, DeRose eliminates the need to generate many small documents with hyperlinks between them. [See DeRose, col. 21, lines 45-47]. How *collections* of the individual documents are arranged and organized on a "Web site" is irrelevant to DeRose, because DeRose is only focused on how portions of the contents of a single document can be easily retrieved in any given session. In other words, DeRose neither discloses nor suggests a method for generating a table of contents for the collection of individual documents, because DeRose is focused on how to retrieve portions of the contents of an individual document. Contrary to the assertions made by the Examiner, considering the patent as a whole, DeRose is directed to individual documents only. Thus, since DeRose is only focused on the contents of an individual document at a time, DeRose does not disclose or suggest a need to look outside an individual document to retrieve information by either indexing for files or scanning the files.

In response to Applicants' argument regarding the rejection of claims 6, 29, and 46 that DeRose does not teach the step of "indexing each file and a first level of each book of



a predetermined folder for files of a first type," the Examiner has asserted that DeRose teaches that each Web site comprises a table of contents (index) of related HTML documents, thus indexing of each file and a first level of each book (documents) for files of a first type (related files or documents) is inherently included in DeRose's teaching of "Web site." [See Final Office Action, Paper No. 8, section 5, page 5]. The Examiner further asserts that each of the files is further scanned for HTML meta-tags for adding data to a table of contents which lists elements of the individual file. Each element is asserted to be further scanned for the element's table of contents.

Contrary to the Examiner's assertions, DeRose does not teach that a "Web site" generates a table of contents after indexing for files of a first type. At best, DeRose discloses that a "Web site" contains documents that can be organized through a table of contents. However, it is respectfully submitted that a table of contents associated with a Web site is not generated dynamically. Rather, it is created by the Web site designer and must be updated manually when information changes. Thus, if a link in the table of contents changes and the table of contents is not manually updated, the entry in the table of contents will cease to function. This is precisely one of the problems that the present invention attempts to resolve. Because the table of contents for a Web site must be manually updated, contrary to the Examiner's assertions, DeRose does not disclose or suggest a method by which a Web site indexes for files of a first type as a step in a method for generating a table of contents.

In addition, DeRose does not disclose or suggest any method for scanning separate, individual files for HTML meta-tags in order to determine data to be added to a table of contents. As discussed previously, the table of contents that are created by DeRose are for the contents of an individual document. Since DeRose is concerned with allowing a user to access portions contained within an individual file, DeRose does not disclose or suggest any method for scanning outside each of the individual files to gather information to generate a table of contents for the collections of individual files. Thus, even if DeRose discloses "Web sites" that comprise a table of contents, since the table of contents for a Web site are manually updated and DeRose is concerned with the contents of individual

files and not collections of files, DeRose does not disclose or suggest that the table of contents is dynamically generated either after indexing each file for *files* of a first type or after scanning the *files* of a first type for at least one HTML meta-tag of a predetermined type in order to determine the data to be added to a table of contents.

Therefore, because DeRose does not disclose or suggest several features of claims 6, 29, 46, and 57, DeRose does not render the subject matter of claims 6-7, 9, 13-14, 29-30, 32, 36-37, 46-47, 49, 53-54, 57-58 unpatentable.

D. Claims 8, 31, 48

Claim 8 of the present application recites that the files of a first type are text files having a predetermined creator designation. Claims 31 and 48 recites similar steps or features. DeRose does not disclose or suggest that the files can be text files with a creator designation.

Column 15, lines 23-25 of DeRose discloses the use of additional style elements in the electronic document. [See DeRose, col. 14, lines 40-44: "The header information is defined by style definition for a "#header" style in a style sheet for the given document type definition of the selected SGML document. This "#header" style should not be confused with a <header> element, but is rather a reserved style name."] These style elements are used to place additional information either before (the "#header") or after (the "#footer") sections of the document. In other words, style elements format how information contained within a particular document is laid out and displayed to a user.

A creator designation, however, is a characteristic of the document itself — it is a specification of which program created the given file. Since a "style element" does not identify the source or type of a document, DeRose does not disclose or suggest the use of creator designations. Accordingly, DeRose does not disclose or suggest the particular aspect of the present invention recited in claims 8, 31, and 48.

In response to Applicants' argument regarding the rejection of claims 8, 31, and 48 that DeRose does not teach a "creator designator," the Examiner asserts that Applicants appear to read limitations from the specification into the claims. [See Final Office Action,

Paper No. 8, section 5, pages 5-6]. It is further asserted by the Examiner that the limitation "creator designation," by itself, cannot be interpreted as having the function of defining a file's type or class, because its function of defining the file's type or class is not recited in the claim. Thus, it is contended by the Examiner that DeRose's teaching of the content appended to the document reads on the claimed "creator designation." It is respectfully submitted that the Examiner interprets the claim terminology in a manner that is not consistent with its commonly understood meaning.

According to M.P.E.P. § 2111, during examination of the pending claims, "the broadest reasonable interpretation of the claims must . . . be consistent with the interpretation that those skilled in the art would reach." [M.P.E.P. § 2111]. This means that "the words of the claim must be given their plain meaning unless applicant has provided a clear definition in the specification." [M.P.E.P. § 2111.01]. Accordingly, pending claims cannot be interpreted so broadly that the meaning given to the terms used in the claims is no longer consistent with the ordinary meaning of those terms. Therefore, in asserting that DeRose's teaching of the content appended to the document reads on the claimed "creator designation," it is respectfully submitted that the claim terminology is being read more broadly than is consistent with the ordinary meaning of the term "creator designation."

A creator designation, according to its "plain meaning," is a characteristic of the document itself. A creator designation designates the type of file the document is or the class of files to which the document belongs. In other words, the creator designation is a specification of which program created the given file. For example, creator designations are used in the Macintosh® operating system to associate files with executable programs (*e.g.*, Microsoft Word documents with the Microsoft Word program). As another example, the Windows® operating system uses the three-digit suffix of the 8.3 file naming convention to perform the same function (*e.g.*, ".doc" associates Microsoft Word documents with the Microsoft Word program). [See the present application, page 7, lines 12-18]. In contrast, style elements format how information contained within a particular document is laid out and displayed to a user.

By interpreting "creator designation" — which specifies the type of a file or the class to which a file belongs — so broadly that it reads on style elements — which are used to format information contained within a document for display — it is respectfully submitted that the term "creator designation" is being improperly interpreted beyond the term's plain meaning. Interpreting the term "creator designation" to mean "style element" renders the term inconsistent with its plain meaning, since a "style element" does not identify the source or type of a document. Therefore, contrary to the Examiner's assertions, it is respectfully submitted that DeRose's teaching of the content appended to the document does not read on the claimed "creator designation."

It should also be noted that, because DeRose deals with individual documents, DeRose has no need to use creator designations. In other words, since DeRose is used to organize the information contained within an individual document, DeRose does not teach a method for scanning numerous, individual files to locate information outside an individual document. Since DeRose does not teach a method of scanning numerous, individual files, DeRose does not require a method of uniquely identifying these files to expedite file identification. Thus, contrary to the Examiner's assertions, DeRose does not teach the use of a "creator designation" which defines a file's type or source.

Thus, DeRose does not disclose or suggest that the files can be text files with a creator designation. Consequently, DeRose does not render the subject matter of claims 8, 31, and 48 unpatentable.

Hence, the subject matter of claims 8, 31, and 48 is separately patentable for this additional reason.

E. Claims 10, 33, and 50

Claim 10 of the present application recites a method for generating a table of contents based on a multi-file system. Claims 33 and 50 recites similar steps or features. DeRose does not disclose or suggest a method for generating a table of contents based on a multi-file system.

As discussed previously, because DeRose is focused on the contents of an individual document and not collections of files, techniques for handling multi-file systems are simply outside the teachings of DeRose. [See DeRose, col. 4, lines 34-39]. Thus, DeRose does not disclose or suggest the particular aspect of the present invention recited in claims 10, 33, and 50.

In response to Applicants' argument regarding the rejection of claims 10, 33, and 50 that DeRose fails to teach the generation of a table of contents of a computerized help system comprised of numerous files, the Examiner has asserted that DeRose teaches the retrieval of numerous individual files (documents) through the WWW. [See Final Office Action, Paper No. 8, section 5, page 5]. The Examiner further asserts that for each file, a table of contents for the file is generated to help in downloading a specific portion of the file instead of the entire file, thus avoiding processing waste.

As discussed previously, the objective of DeRose is to provide a mechanism for accessing only a portion of the information contained within a large electronically published document to avoid the document management problems associated with maintaining collections of small document fragments. Contrary to the Examiner's assertions, how collections of individual documents are organized, on a "Web site" for example, is not a concern of DeRose. As a result, DeRose does not teach the generation of a table of contents for the numerous individual documents. Since DeRose specifically teaches a method for breaking down a single document into smaller parts to facilitate retrieval of portions of that document, it is respectfully submitted that DeRose fails to teach the generation of a table of contents of a computerized help system comprised of numerous individual files.

Therefore, DeRose does not disclose or suggest a method for generating a table of contents based on a multi-file system. Consequently, DeRose does not render the subject matter of claims 10, 33 and 50 unpatentable.

Hence, the subject matter of claims 10, 33, and 50 is separately patentable for this additional reason.

F. Claims 11, 34, and 51

Claim 11 of the present application recites that the table of contents for each book is formatted using an HTML template. Claims 34 and 51 recite similar steps or features. In response to Applicants' argument regarding the rejection of claims 11, 34, and 51 that DeRose does not teach the use of an HTML template for generating the table of contents, the Examiner has asserted that DeRose discloses at column 19, lines 4-22 and in Figure 9 (#160) the HTML template used for generating the table of contents. [See Final Office Action, Paper No. 8, section 5, page 6]. Contrary to the Examiner's assertions, DeRose does not indicate the use of templates to format the table of contents, but merely the use of a type of style definition which specifies the various titles in the document.

As discussed by DeRose, the table of contents display retrieves the title information for the siblings of a selected element, the children of the selected element, and the siblings of ancestors of the selected element. The resulting table of contents document includes the printed titles of each section as hypertext links to those sections. However, contrary to the Examiner's assertions, DeRose does not disclose the step of formatting the table of contents using an HTML template. A template is a file or form that defines the layout of a document. Although according to DeRose the table of contents is displayed to a user, contrary to the Examiner's assertions, DeRose does not disclose or suggest that the displayed table of contents is laid out according to a file or form (i.e., a template) that defines the layout of DeRose's table of contents document. DeRose uses style elements (i.e., title bearers) to generate the table of contents, but these style elements do not define the layout of DeRose's table of contents. [See DeRose, col. 18, lines 33-34].

Hence, the subject matter of claims 11, 34, and 51 is separately patentable for this additional reason.

G. Claims 12, 35, and 52

Claim 12 of the present application recites that the formatted table of contents is provided upon user request. Claims 35 and 52 recite similar steps or features. It is

respectfully submitted that DeRose does not disclose or suggest that the table of contents is provided upon user request.

DeRose generates a table of contents, so long as the "title bearer" style definitions are present within the document. No determination is made by DeRose as to whether or not a table of contents *should* be generated — if the title bearer style definitions exist, DeRose generates a table of contents. [See DeRose, col. 18, line 59 - col. 19, line 3]. Depending on the size of the portion of the document selected by the user, either the document portion itself is sent and displayed to the user, or, if the portion is too large, a table of contents is sent and displayed to the user instead. [See DeRose, col. 13, lines 18-23]. Thus, DeRose does not disclose or suggest the particular aspect of the present invention recited in claims 12, 35, and 52.

In response to the Applicants' argument regarding the rejection of claims 12, 35, and 52 that DeRose does not implicitly include that the table of contents is provided upon user request, the Examiner has asserted that the table of contents is provided in response to the user selection of the book which sends a signal (implicitly included) requesting the displaying of the table of contents. [See Final Office Action, Paper No. 8, section 5, page 6]. Contrary to the Examiner's assertions, as discussed previously, no mechanism for determining whether a table of contents should be generated upon user request (as that contemplated by the present invention) is taught by DeRose. Rather, depending on the size of the portion of the document selected by the user, either the document portion itself is sent and displayed to the user, or, if the portion is too large, a table of contents is sent and displayed to the user instead. [See DeRose, col. 13, lines 18-23].

In addition, DeRose generates a table of contents to facilitate user input. DeRose discloses a mechanism for determining whether a table of contents needs to be *displayed*, and displays the table of contents to facilitate user input to begin access to portions of the document. As discussed by DeRose, "[a] variety of mechanisms may [be] used for obtaining from a user a desired starting point from which rendering of a document may begin. One mechanism is a table of contents from which a section of a document may be selected . . . ." [DeRose, col. 11, lines 38-41]. In other words, according to DeRose, if

the document is large enough then the table of contents is displayed first, in response to which a user inputs a request to view a selected portion of a document. As discussed by DeRose, "a table of contents is generated for a document upon receipt of an indication of an element within that document." [DeRose, col. 18, lines 57-59]. The "element" referred to is not a user input, but instead the title bearer style element discussed previously. Therefore, contrary to the Examiner's assertions, DeRose does not disclose that the table of contents is provided in response to the user selection of the book which sends a signal requesting the displaying of the table of contents. Rather, DeRose generates a table of contents based on embedded style elements and provides the table of contents *before* a user makes a request in order to facilitate such requests.

Since DeRose does not disclose or suggest that the table of contents is provided upon user request, DeRose does not render the subject matter of claims 12, 35, and 52 unpatentable.

Hence, the subject matter of claims 12, 35, and 52 is patentable for this additional reason.

#### H. Claims 15, 38, and 59

Claim 15 of the present application recites the determination of whether a table of contents is to be generated. Claims 38 and 59 recite similar steps and features. DeRose does not disclose or suggest the determination of whether a table of contents is to be generated.

As discussed previously, DeRose discloses a mechanism for determining whether a table of contents needs to be *displayed*. [See DeRose, col. 13, lines 18-21]. Consequently, since DeRose does not disclose or suggest a mechanism for determining whether a table of contents needs be *generated* (as that contemplated by the present invention), DeRose does not disclose or suggest the particular aspect of the present invention recited in claims 15, 38, and 59.

In response to the Applicants' argument regarding the rejection of claims 15, 38, and 59 that DeRose does not contemplate the determination of whether a table of contents



is to be generated, the Examiner has asserted that a table of contents is generated and displayed based upon user input and is generated if it is determined that the size of a document is too large. [See Final Office Action, Paper No. 8, section 5, page 6]. Contrary to the Examiner's assertions, as discussed previously, DeRose makes a determination as to whether or not to *display* the generated table of contents. In addition, as discussed previously, a table of contents is generated in order to facilitate user input. Consequently, no such mechanism for determining whether a table of contents needs to be *generated* (as that contemplated by the present invention) is taught by DeRose.

Since the DeRose does not disclose or suggest the determination of whether a table of contents is to be generated, DeRose does not render the subject matter of claims 15, 38, and 59 unpatentable.

Hence, the subject matter of claims 15, 38, and 59 is patentable for this additional reason.

I. Claims 55 and 56

Claim 55 of the present application recites the generation of a table of contents only if a template exists. Claim 56 recites a similar feature. DeRose does not disclose or suggest the generation of a table of contents only if a template exists.

In response to Applicants' argument regarding the rejection of claims 55 and 56 that DeRose does not teach the generation of a table of contents only if a template exists, the Examiner has asserted that a template is disclosed by DeRose and that the table of contents is created based on the template and, thus, the determining whether the template exists is implicitly included in DeRose. [See Final Office Action, Paper No. 8, section 5, page 7].

As was discussed previously, contrary to the Examiner's assertions, DeRose does not disclose or suggest the use of templates for creating the table of contents, because DeRose does not disclose or suggest the use of a file or form that defines the layout of a document (i.e., the table of contents). As noted previously, DeRose uses a type of style definition which specifies what are the various titles in the document. These titles are then retrieved and placed in the table of contents. However, DeRose does not disclose, either

implicitly or explicitly, that a file or form is used to control the placement and format of the titles within the table of contents.

Furthermore, DeRose does not teach the generation of a table of contents only if the template exists. Rather, DeRose will only generate a table of contents if the electronic document contains the necessary title bearer style elements. [See DeRose, col. 18, lines 33-34]. Thus, contrary to the Examiner's assertions, since DeRose does not disclose or suggest the use of templates, DeRose does not disclose or suggest — either implicitly or explicitly — the generation of a table of contents only when the template exists.

Hence, the subject matter of claims 55 and 56 is patentable for this additional reason.

J. Claims 2-5, 16-23, 25-28, 39-45, and 60-61

As discussed earlier, DeRose specifically teaches away from a method of creating a table of contents based upon numerous, individual information files (*i.e.*, a multi-file system) as being contrary to the stated objective of the invention of DeRose. [See DeRose, col. 4, lines 34-39]. Walls, however, teaches a method of searching one or more files within a file system to provide the user with an updated index. Since DeRose specifically teaches away from the use of multiple files, one of ordinary skill in the art would not be motivated to modify its system in a manner which goes against the express teachings of DeRose. Therefore, since the teachings of DeRose and Walls are directed to disparate objectives, and in fact are contrary to each other, it would not have been obvious to one of ordinary skill in the art, at the time the invention was made, to combine the teachings of DeRose and Walls (or DeRose and any other teaching which suggests the use of a multi-file system).

With regards to claim 60, the Examiner acknowledges that DeRose fails to teach the step of determining if the table of contents needs to be generated if the table of contents fails to either exist or be up-to-date. As discussed previously, DeRose does not contemplate the determination of whether a table of contents is to be generated. Rather, DeRose discloses a mechanism for determining whether a table of contents needs to be

*displayed*. Therefore, since DeRose does not disclose or suggest the determination of whether a table of contents is to be generated, it would not have been obvious to one of ordinary skill in the art, at the time the invention was made, to implement the determining if the table of contents needs to be generated if it fails to either exist or be up-to-date.

In response to Applicants' arguments regarding the rejection of claims 2-5, 16-28, and 39-45, it is again asserted by the Examiner that it would have been obvious to one of ordinary skill in the art to combine Walls' teaching of creating the up-to-date table of contents with DeRose. [See Final Office Action, Paper No. 8, section 5, page 7]. The motivation for such a combination is asserted by the Examiner to be the provision of an up-to-date table of contents. Contrary to the Examiner's assertions, as discussed previously, it is respectfully submitted that it would not have been obvious to combine the teachings of Walls with the teachings of DeRose, since one of ordinary skill in the art would not be motivated to modify the DeRose system in a manner which goes against the express teachings of DeRose.

Additionally, the Examiner has asserted that Walls' index is a table of headers in a set of files displayed to the user in a graphical representation of a directory or tree of a file system, and is thus a table of contents. Contrary to the Examiner's assertions, Walls' index is not a table of headers in a set of files. Rather, Walls' index is comprised of the information that is contained *within* the file, not the files themselves.

According to Walls, to reduce the amount of time needed to create the index, only the header of each file is searched for index-organizing elements that characterize subjects of the information contained within the file system and the corresponding location elements that identify which of the files contains the subject characterized by the corresponding index-organizing elements. [See Walls, col. 3, lines 55-59]. It should be noted that the location elements do not describe where a particular file is located, but rather specify to the user the name of the file in which the corresponding information is located. Once the continuous indexer disclosed by Walls extracts from the files the index-organizing elements and descriptive elements, the indexer organizes these elements in a predetermined format, such as in alphabetical order. [See, e.g., Walls, figure 8]. Thus, Walls does not list the

headers of a set of files, but instead lists the contents of the headers of a set of files. Consequently, the index that is created by Walls enables a user to efficiently search files to locate a particular item of information. In contrast, a table of contents, according to the present invention, is designed to provide the user with a high-level overview of the structure and layout of a file or collection of files in a computer system. Therefore, contrary to the Examiner's assertions, the index, as contemplated in Walls, is not the same as a table of contents, as contemplated by the present invention.

Since one of ordinary skill in the art would not be motivated to combine the teachings of DeRose and Walls, the combination of DeRose and Walls does not render the subject matter of claims 2-5, 16-23, 25-28, 39-45, and 60-61 obvious.

Hence, the subject matter of claims 2-5, 16-23, 25-28, 39-45, and 60-61 is patentable for this additional reason.

IX. Conclusion

For the reasons presented above, the rejections of the claims are not properly founded in the statute and should be reversed.

Respectfully submitted,

BURNS, DOANE, SWECKER & MATHIS, L.L.P.

By: Andrew J. Bateman  
Andrew J. Bateman  
Registration No. 45,573

P.O. Box 1404  
Alexandria, Virginia 22313-1404  
(703) 836-6620

Date: August 17, 2001

## APPENDIX

### The Appealed Claims

NOTE: Claims 1 and 24 are not being appealed, but are included in this Appendix because claims 2 and 25, which are being appealed, depend from them.

1. (Amended) A method for displaying information to a user of a computer system, comprising the steps of:

activating a computerized information system;

dynamically generating a table of contents in response to said activation; and

displaying said dynamically generated table of contents;

wherein said computer system displays an up-to-date listing of available information.

2. (Amended) The method according to claim 1 wherein the step of dynamically generating comprises the steps of:

indexing each file and book of a predetermined folder for files of a first type; and

scanning said files of a first type for at least one HTML meta-tag of a

predetermined type in order to determine data to be added to said table of contents.

3. The method according to claim 2 wherein said files of a first type are text files having a predetermined creator designation.

4. The method according to claim 2 wherein said at least one HTML meta-tag is a meta-tag for providing a title.

5. The method according to claim 2 wherein said predetermined folder is a folder that is uniquely associated with a help system.

6. (Amended) A method for providing a dynamically generated table of contents upon activation of a computerized information help system, comprising the steps of:

indexing each file and a first level of each book of a predetermined folder for files of a first type;

scanning said files of a first type for at least one HTML meta-tag of a predetermined type in order to determine first data to be added to a first table of contents;

generating said first table of contents using said first data;

formatting said first table of contents using a template; and

displaying said formatted first table of contents.

7. The method according to claim 6 wherein said predetermined folder is a folder that is uniquely associated with a help system.

8. The method according to claim 6 wherein said files of a first type are text files having a predetermined creator designation.

9. The method according to claim 6 wherein said at least one HTML meta-tag is a meta-tag for providing a title.

10. (Amended) The method according to claim 6, further comprising the steps of:  
indexing each chapter within each book of said predetermined folder for said files of a first type;  
scanning said files of a first type of each book for at least one HTML meta-tag of a predetermined type in order to determine second data to be added to a second table of contents for each book; and  
dynamically generating said second table of contents for each book using said second data.
11. (Amended) The method according to claim 10 further comprising the step of formatting said second table of contents for each book using an HTML template.
12. The method according to claim 11 further comprising the step of providing said formatted second table of contents for each book in response to a request.
13. (Amended) The method according to claim 6 wherein said formatted first table of contents comprises at least one link to a location of a file on the computer system.
14. The method according to claim 6 wherein said formatted first table of contents comprises at least one link to a web page on the Internet.

15. (Amended) The method according to claim 6, further comprising the step of:  
determining, in response to the activation of said information help system, whether  
a table of contents needs to be generated.

16. (Amended) The method according to claim 15 wherein the step of determining  
further comprises the steps of:  
determining whether a table of contents for said help system already exists;  
determining, if said table of contents already exists, whether said table of contents  
is up-to-date; and  
determining that said table of contents needs to be generated when said table of  
contents fails to either exist or be up-to-date.

17. (Amended) The method according to claim 16 wherein the step of determining  
whether the table of contents is up-to-date comprises the steps of:  
comparing a modification date of files to be used in the generation of said first table  
of contents and a date in which said table of contents was last generated; and  
determining that said table of contents is not up-to-date when said modification date  
is more recent than said generation date.

18. (Amended) A method for displaying a dynamically generated table of contents of a  
computerized information help system, comprising the steps of:  
activating said help system;  
determining whether a table of contents for said help system already exists;



determining, if said table of contents already exists, whether said table of contents is up-to-date;

dynamically generating a table of contents when said table of contents fails to either exist or be up-to-date; and

displaying said dynamically generated table of contents.

19. (Amended) The method according to claim 18 wherein the step of determining whether the table of contents is up-to-date comprises the steps of:

comparing a modification date of files to be used in the generation of said table of contents and a date in which the table of contents was last generated; and

determining that said table of contents is not up-to-date when said modification date is more recent than said generation date.

20. (Amended) The method according to claim 18 wherein the step of dynamically generating comprises the steps of:

indexing each file and book of a predetermined folder for files of a first type; and

scanning said files of a first type for at least one HTML meta-tag of a predetermined type in order to determine data to be added to said table of contents.

21. The method according to claim 20 wherein said files of a first type are text files having a predetermined creator designation.

22. The method according to claim 20 wherein said at least one HTML meta-tag is a meta-tag for providing a title.

23. The method according to claim 20 wherein said predetermined folder is a folder that is uniquely associated with a help system.

24. (Amended) An apparatus for displaying information to a user of a computer system, comprising:

a viewer for dynamically generating a list of books currently available to the user;

and

a display device for displaying said list of currently available books;

wherein said viewer dynamically generates said list of books in response to a request.

25. (Amended) The apparatus according to claim 24 wherein said viewer generates said list of books by indexing each file and book of a predetermined folder for files of a first type and scanning said files of a first type for at least one HTML meta-tag of a predetermined type in order to determine data to be added to said list of books.

26. The apparatus according to claim 25 wherein said files of a first type are text files having a predetermined creator designation.

27. The apparatus according to claim 25 wherein said at least one HTML meta-tag is a meta-tag for providing a title.

28. The apparatus according to claim 25 wherein said predetermined folder is a folder that is uniquely associated with a help system.

29. (Amended) A system for displaying a dynamically generated table of contents upon activation of a computerized information help system, comprising:

means for indexing each file and a first level of each book of a predetermined folder for files of a first type;

means for scanning said files of a first type for at least one HTML meta-tag of a predetermined type in order to determine first data to be added to a first table of contents;

means for generating said first table of contents using said first data;

means for formatting said first table of contents using a template; and

means for displaying said formatted first table of contents.

30. The system according to claim 29 wherein said predetermined folder is a folder that is uniquely associated with a help system.

31. The system according to claim 29 wherein said files of a first type are text files having a predetermined creator designation.

32. The system according to claim 29 wherein said at least one HTML meta-tag is a meta-tag for providing a title.

33. (Amended) The system according to claim 29, further comprising:

means for indexing each chapter within each book of said predetermined folder for files of a first type;

means for scanning said files of a first type of each book for at least one HTML meta-tag of a predetermined type in order to determine second data to be added to a second table of contents for each book; and

means for dynamically generating said second table of contents for each book using said second data.

34. (Amended) The system according to claim 33 further comprising means for formatting said second table of contents for each book using an HTML template.

35. The system according to claim 34 further comprising means for displaying said formatted second table of contents for each book in response to a request.

36. The system according to claim 29 wherein said formatted first table comprises at least one link to a location of a file on the computer system.

37. The system according to claim 29 wherein said formatted first table of contents comprises at least one link to a web page on the Internet.

38. (Amended) The system according to claim 29, further comprising:

means for determining, in response to the activation of said information help system, whether a table of contents needs to be generated.

39. (Amended) The system according to claim 38 wherein the means for determining further comprises:

means for determining whether said table of contents for said help system already exists;

means for determining, if said table of contents already exists, whether said formatted first table of contents is up-to-date; and

means for determining that said table of contents needs to be generated when said table of contents fails to either exist or be up-to-date.

40. (Amended) The system according to claim 39 wherein the means for determining whether the table of contents is up-to-date comprises:

means for comparing a modification date of files to be used in the generation of said first table of contents and a date in which said table of contents was last generated; and

means for determining that said table of contents is not up-to-date when said modification date is more recent than said generation date.

41. A computer-readable medium containing a program which performs the steps of:

activating a computerized information system;

dynamically generating a table of contents in response to said activation; and

displaying said dynamically generated table of contents;  
wherein said dynamically generated table of contents is an up-to-date listing of  
available information.

42. (Amended) The computer-readable medium according to claim 41 wherein [said]  
the step of dynamically generating comprises the steps of:

indexing each file and book of a predetermined folder for files of a first type; and  
scanning said files of a first type for at least one HTML meta-tag of a  
predetermined type in order to determine data to be added to said table of contents.

43. The computer-readable medium according to claim 42 wherein said files of a first  
type are text files having a predetermined creator designation.

44. The computer-readable medium according to claim 42 wherein said at least one  
HTML meta-tag is a meta-tag for providing a title.

45. The computer-readable medium according to claim 42 wherein said predetermined  
folder is a folder that is uniquely associated with a help system.

46. (Amended) A computer-readable medium containing a program which performs the  
steps of:

indexing each file and a first level of each book of a predetermined folder for files  
of a first type;

scanning said files of a first type for at least one HTML meta-tag of a predetermined type in order to determine first data to be added to a first table of contents;

generating said first table of contents using said first data;

formatting said first table of contents using a template; and

displaying said formatted first table of contents.

47. The computer-readable medium according to claim 46 wherein said predetermined folder is a folder that is uniquely associated with a help system.

48. The computer-readable medium according to claim 46 wherein said files of a first type are text files having a predetermined creator designation.

49. The computer-readable medium according to claim 46 wherein said at least one HTML meta-tag is a meta-tag for providing a title.

50. (Amended) The computer-readable medium according to claim 46, further comprising the steps of:

indexing each chapter within each book of said predetermined folder for files of a first type;

scanning said files of a first type of each book for at least one HTML meta-tag of a predetermined type in order to determine second data to be added to a second table of contents for each book; and

dynamically generating said second table of contents for each book using said second data.

51. (Amended) The computer-readable medium according to claim 50 further comprising the step of formatting said second table of contents for each book using an HTML template.

52. The computer-readable medium according to claim 51 further comprising the step of providing said formatted second table of contents for each book in response to a request.

53. (Amended) The computer-readable medium according to claim 46 wherein said formatted first table of contents comprises at least one link to a location of a file on a computer system.

54. The computer-readable medium according to claim 46 wherein said formatted first table of contents comprises at least one link to a web page on the Internet.

55. (Amended) The method according to claim 15 wherein the step of determining further comprises the steps of:

determining whether a template exists; and

determining that said table of contents needs to be generated only when said template exists.



56. (Amended) The system according to claim 38 wherein [said] the means for determining further comprises:

means for determining whether a template exists; and

means for determining that said table of contents needs to be generated only when said template exists.

57. (New) A method for facilitating a user's access to a computerized help system of a type comprising multiple files of information, comprising the steps of:

scanning plural files that are accessible by a computer to identify files of a first type;

analyzing said identified files of a first type to locate at least one meta-tag of a predetermined type in order to determine first data to be added to a table of contents;

generating said table of contents using said first data;

formatting said table of contents using a template; and

displaying said formatted table of contents.

58. (New) The method of claim 57 wherein the files that are scanned are stored within a predetermined folder in a file system.

59. (New) The method according to claim 57, further comprising the step of:

determining, in response to the activation of said help system, whether a table of contents needs to be generated.

60. (New) The method according to claim 59 wherein the step of determining further comprises the steps of:

determining whether a table of contents for said help system already exists;

determining, if said table of contents already exists, whether said table of contents is up-to-date; and

determining that said table of contents needs to be generated when said table of contents fails to either exist or be up-to-date.

61. (New) The method according to claim 60 wherein the step of determining whether the table of contents is up-to-date comprises the steps of:

comparing a modification date of files to be used in the generation of said table of contents and a date in which said table of contents was last generated; and

determining that said table of contents is not up-to-date when said modification date is more recent than said generation date.

## TABLE OF CONTENTS

	<u>PAGE</u>
I. <u>Real Party in Interest</u> . . . . .	1
II. <u>Related Appeals and Interferences</u> . . . . .	1
III. <u>Status of Claims</u> . . . . .	1
IV. <u>Status of Amendments</u> . . . . .	1
V. <u>Summary of the Invention</u> . . . . .	2
VI. <u>The Issues</u> . . . . .	4
VII. <u>Grouping of Claims</u> . . . . .	4
VIII. <u>Argument</u> . . . . .	4
IX. <u>Conclusion</u> . . . . .	19